UNITED STATES PATENT AND TRADEMARK OFFICE

1.1

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/706,656	11/12/2003	Hong Gui	11166 (NCR.0114US)	8622

7590 12/27/2006

John D. Cowart
NCR Corporation
Law Department IP WHQ-4W
1700 S. Patterson Blvd.
Dayton, OH 45479

EXAMINER	
TIMBLIN, ROBERT M	
ART UNIT	PAPER NUMBER
2167	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	12/27/2006	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	**Application No.** 10/706,656	**Applicant(s)** GUI ET AL.	
	Examiner Robert M. Timblin	**Art Unit** 2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 September 2006*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-28* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-28* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

This office action corresponds to application 10/706,656 and the Applicants' responses filed 9/19/2006. Claims1-28 remain pending in this application. Responses to arguments begin on page 22 of this document.

Response to Remarks

The Examiner thankfully accepts the Applicants' explanation of the first grouping set of claim 1. The objection to this claim for that reason is therefore removed.

Response to Amendment

The Examiner acknowledges the addition of the word being to the last steps of claims 20 and 27. No other amendments have been made.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent

Claims 1-5, 13-16, and 22-24 are rejected under 35 U.S.C. 102(a) as being anticipated by Cochrane et al. (U.S. 2003/0093407) (hereinafter Cochrane).

Regarding claim 1, a database system comprising:

a storage (Figure 1, element 104) to store a view containing results of a cube based operation on at least one base table, the view containing a first result set for a group-by on a second grouping set (paragraphs 0036 and 0038); and

a controller (element 118, paragraph 0024), in response to a change to the at least one base table, to:

update the first result set by computing a change to the first result set based on a change in the at least one base table; and update the second result set by computing a change to the second result set based on the change to the first result set (paragraphs 0035 and 0038; Cochrane discloses an update to a self-maintainable materialized view in paragraph 0035, and in paragraph 0038 Cochrane discloses wherein the statement that defines the materialized view contains a ROLLUP operation. A rollup operation contains multiple grouping sets (the first being the equivalent of a first result set), the first displaying summary data of all the columns to be processed, the second with summary data for all the columns of the first, minus one column that is aggregated, the third with yet another column aggregated and so on until all columns are aggregated.

Paragraph 0035 discloses a method in which changes (the equivalent of updates) can be maintained incrementally by processing the updated rows and visiting the underlying tables [of the materialized view query] to recompute some portions of the materialized view query. Therefore, an update affecting a second result set would be calculated by visiting the already materialized view query to obtain pre-computed rows of data (the first result set) in order to compute an update to a second result set, meeting the limitation computing a change to the second result set based on the change to the first result set.).

Regarding claim 2, the database system of claim 1, wherein the first grouping set has a first number of grouping attributes, and the second grouping set has a second number of grouping attributes, the first number being greater than the second number (paragraphs 0035 and 0038; Cochrane discloses an update to a self-maintainable materialized view in paragraph 0035, and in paragraph 0038 Cochrane discloses wherein the statement that defines the materialized view contains a ROLLUP operation. A rollup operation contains multiple grouping sets (the first being the equivalent of a first result set), the first displaying summary data of all the columns to be processed, the second (the equivalent of the second result set) with summary data for all the columns of the first, minus one column that is aggregated, the third with yet another column aggregated and so on until all columns are aggregated).

Regarding claim 3, the database system of claim 2, wherein the view further contains a third result set for a group-by on a third grouping set having a third number of grouping attributes, the third number less than the second number, the controller to further update the third result set by computing a change to the third result set based on the change to the second result set (paragraphs 0035 and 0038; Cochrane discloses an update to a self-maintainable materialized view in paragraph 0035, and in paragraph 0038 Cochrane discloses wherein the statement that defines the materialized view contains a ROLLUP operation. A rollup operation contains multiple grouping sets (Paragraph 0035 discloses a method in which changes (the equivalent of updates) can be maintained incrementally by processing the updated rows and visiting the underlying tables [of the materialized view query] to recompute some portions of the materialized view query. Therefore, an update affecting a third result set would be calculated by visiting the already materialized view query to obtain pre-computed rows of data (a second result set) in order to compute an update to a third result set, meeting the limitation computing a change to the third result set based on the change to the second result set.).

Regarding claim 4, the database system of claim 1, wherein the view contains results of a group-by cube operation (paragraph 0038).

Regarding claim 5, the database system of claim 1, wherein the view contains results of a group-by partial cube operation (paragraph 0038; a group by grouping sets is the equivalent of a group-by partial cube operation).

Regarding claim 13, a method for use in a database system, comprising: storing a view containing results of a cube-based operation on at least one base table (Figure 1, element 104; the base table is disclosed in paragraph 0035), the view containing result sets for group-bys on respective grouping sets (paragraph 0038); updating a first result set by computing a change to the first result set based on a change in the at least one base table and updating a second result set by computing a change to the second result set based on the change to the first result set (paragraphs 0035 and 0038; Cochrane discloses an update to a self-maintainable materialized view in paragraph 0035, and in paragraph 0038 Cochrane discloses wherein the statement that defines the materialized view contains a ROLLUP operation). A rollup operation contains multiple grouping sets (the first being the equivalent of a first result set), the first displaying summary data of all the columns to be processed, the second with summary data for all the columns of the first, minus one column that is aggregated, the third with yet another column aggregated and so on until all columns are aggregated. Paragraph 0035 discloses a method in which changes (the equivalent of updates) can be maintained incrementally by processing the updated rows and visiting the underlying tables [of the materialized view query] to recompute some portions of the materialized view query. Therefore, an update affecting a second result set would be calculated by visiting the already materialized view query to obtain pre-computed rows of data (the first result set) in order to compute an update to a second result set, meeting the limitation computing a change to the second result set based on the change to the first result set.).

Regarding claim 14, the method of claim 13, wherein updating the first result set comprises updating the first result set for the group-by on a first grouping set that has a greater number of columns than a second grouping set corresponding to the second result set (paragraphs 0035 and 0038; Cochrane discloses an update to a self-maintainable materialized view in paragraph 0035, and in paragraph 0038 Cochrane discloses wherein the statement that defines the materialized view contains a ROLLUP operation. A rollup operation contains multiple grouping sets (the first being the equivalent of a first result set), the first displaying summary data of all the columns to be processed, the second (the equivalent of the second result set) with summary data for all the columns of the first, minus one column that is aggregated, the third with yet another column aggregated and so on until all columns are aggregated).

Regarding claim 15, the method of claim 13, further comprising updating a third result set by computing a change to the third result set based on the change to the second result set (Paragraph 0035 discloses a method in which changes (the equivalent of updates) can be maintained incrementally by processing the updated rows and visiting the underlying tables [of the materialized view query] to recompute some portions of the materialized view query.

Therefore, an update affecting a third result set would be calculated by visiting the already materialized view query to obtain pre-computed rows of data (a second result set) in order compute an update to a third result set, meeting the limitation computing a change to the second result set based on the change to the first result set).

Regarding claim 16, the method of claim 15, further comprising updating a fourth result set by computing a change to the fourth result set based on the change to the third result set (Paragraph 0035 discloses a method in which changes (the equivalent of updates) can be maintained incrementally by processing the updated rows and visiting the underlying tables [of the materialized view query] to recompute some portions of the materialized view query. Therefore, an update affecting a fourth result set would be calculated by visiting the already materialized view query to obtain pre-computed rows of data (a third result set) in order to compute an update to a fourth result set, meeting the limitation computing a change to the fourth result set based on the change to the third result set).

Regarding claim 22, an article comprising at least one storage medium containing instructions that when executed cause a database system to:

store a view containing results of a cube-based operation on at least one base table (paragraph 0035), the view containing result sets for group-bys on respective grouping sets (paragraph 0038);

update a first result set by computing a change to the first result set based on a change in the at least one base table; and update a second result set by computing a change to the second result set based on the change to the first result set (paragraphs 0035 and 0038; Cochrane discloses an update to a self-maintainable materialized view in paragraph 0035, and in paragraph 0038 Cochrane discloses wherein the statement that defines the materialized view contains a ROLLUP operation. A rollup operation contains multiple grouping sets (the first being the equivalent of a first result set), the first displaying summary data of all the columns to be processed, the second with summary data for all the columns of the first, minus one column that is aggregated, the third with yet another column aggregated and so on until all columns are aggregated. Paragraph 0035 discloses a method in which changes (the equivalent of updates) can be maintained incrementally by processing the updated rows and visiting the underlying tables [of the materialized view query] to recompute some portions of the materialized view query. Therefore, an update affecting a second result set would be calculated by visiting the already materialized view query to obtain pre-computed rows of data (the first result set) in order to compute an update to a second result set, meeting the limitation computing a change to the second result set based on the change to the first result set).

Regarding claim 23, the article of claim 22, wherein updating the first result set comprises updating the first result set for the group-by on a first grouping set that has a greater number of columns than a second grouping set corresponding to the result set (paragraphs 0035 and 0038; Cochrane discloses an update to a self-maintainable materialized view in paragraph 0035, and in paragraph 0038 Cochrane discloses wherein the statement that defines the materialized view contains a ROLLUP operation. A rollup operation contains multiple grouping sets (the first being the equivalent of a first result set), the first displaying summary data of all the columns to be processed, the second (the equivalent of the second result set) with summary data for all the columns of the first, minus one column that is aggregated, the third with yet another column aggregated and so on until all columns are aggregated).

Regarding claim 24, the article of claim 22, wherein the instructions when executed cause the database system to further update a third result set by computing a change to the third result set based on the change to the second result set (paragraphs 0035 and 0038; Cochrane discloses an update to a self- maintainable materialized view in paragraph 0035, and in paragraph 0038 Cochrane discloses wherein the statement that defines the materialized view contains a ROLLUP operation. A rollup operation contains multiple grouping sets (Paragraph 0035 discloses a method in which changes (the equivalent of updates) can be maintained incrementally by processing the updated rows and visiting the underlying tables [of the materialized view query] to recompute some portions of the materialized view query. Therefore, an update affecting a third result set would be calculated by visiting the already materialized view query to obtain pre-computed rows of data (a second result set) in order to compute an update to a third result set, meeting the limitation computing a change to the third result set based on the change to the second result set).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 6-12, 17-21, and 25-28 are rejected under 35 U.S.C. 103(a) as being anticipated by Cochrane et al. (U.S. 2003/0093407) (hereinafter Cochrane) in view of Bellamkonda et al. (U.S. 7,035,843) (hereinafter Bellamkonda).

Claims 6-12, 17-21, and 25-28 are rejected under 35 U.S.C. 103(a) as being anticipated by Cochrane et al. (U.S. 2003/0093407) (hereinafter Cochrane) in view of Bellamkonda et al. (U.S. 7,035,843) (hereinafter Bellamkonda).

Regarding claim 6, the database system of claim 1, Cochrane does not explicitly disclose plural access modules and plural storage modules, the access modules to enable parallel access of data in the plural storage modules.

In the same field of endeavor (database analytical processing systems) Bellamkonda discloses plural access modules and plural storage modules, the access modules to enable parallel access of data in the plural storage modules (column 16, line 65-column 17, line 1; slaves in Bellamkonda are the equivalent of access modules. The locations where the rows that each slave processes are the equivalent of plural storage modules). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of processing of plural access modules and plural storage modules with Cochrane's teachings of a database system with multiple views with dependencies to obtain a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in

Regarding claim 7, Cochrane does not explicitly disclose wherein the controller is adapted to distribute rows in the first result set across the access modules based on a hash of columns of the second grouping set and at least another column that is assigned to a predefined value.

In the same field of endeavor (database analytical processing systems) Bellamkonda discloses wherein the controller is adapted to distribute rows in the first result set across the access modules based on a hash of columns of the second grouping set and at least another column that is assigned to a predefined value (column 17, lines 25-26; see also Figure 4, column 12, lines 4-9, column 13, lines 43-45 and column 14, lines 39-41. Each grouping set of the rollup calculated in Bellamkonda by this method is a result set. Bellamkonda discloses a single field (the equivalent of a column) that is aggregated in the grouping set of the calculations of the next stage (the equivalent of the second grouping set) in column 12, lines 26-28. The single field disclosed is an extra column not considered in the hash of the second result set (the equivalent of the second grouping set), but is considered in the hash of the first result set (the equivalent of the first grouping set)). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of processing of plural access modules and plural storage modules with Cochrane's teachings of a database system with multiple views with dependencies to obtain a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 8, Cochrane discloses a view (paragraph 0035). Cochrane does not explicitly disclose but in the same field of endeavor (database analytical processing systems) Bellamkonda discloses wherein the view contains results of a cube operation specified by a cube function on plural columns, the at least another column being one of the plural columns of the cube function that is not in the second grouping set (column 12, line 16-column 12, line 28;Bellamkonda discloses a rollup operation which is a cube operation, and further discloses a single field (the equivalent of a column) that is aggregated in the grouping set of the calculations of the next stage (the equivalent of the second grouping set)). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of processing of hashing on columns, including an extra column not present in a second grouping set with Cochrane's teachings of a database system with multiple views with dependencies on other views or tables to obtain a system for updating views that processes an extra column in the hashing of a first grouping set than in a second grouping set for the purpose of a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 9, Cochrane does not explicitly disclose but in the same field of endeavor (database analytical processing systems) Bellamkonda discloses wherein the view is distributed across the access modules such that plural portions of the view reside in respective storage modules, and wherein the rows in the first result set are distributed across the access modules according to the hash to enable:

Each access module to locally perform a merge and aggregate operation on the rows of the first result set to produce rows of the second result set (column 16 line 65-column 17 line 1); and

Each access module to locally merge the rows of the second result set into a respective portion of the view without having to first redistribute the rows of the second result set (column 16 line 65-column 17 line 1, slaves (the equivalent of processing modules) work independently to produce their own grouping sets, which can be merged without rearranging rows). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of access modules to locally perform merge and aggregate operations, without having to first redistribute the rows of the second result set with Cochrane's teachings of a database system with multiple views with dependencies on other views or tables to obtain a system for updating views without having to first redistribute the rows of the second result set for the purpose of a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 10, the database system of claim 1, Cochrane does not explicitly disclose but in the same field of endeavor (database analytical processing systems) Bellamkonda discloses wherein the controller is adapted to further:

receive a query specifying a group-by operation (column 17, line 61); and

determine whether an answer for the query specifying the group-by operation can be satisfied from the view (column 17, lines 58-61; a determination to decide if more operations need to be done to generate result sets satisfy a query is the same as determining whether an answer for a query can be satisfied from the view). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of receiving a query and determining whether an answer for the query specifying the group-by operation can be satisfied from the view with Cochrane's teachings of a database system with multiple views with dependencies to obtain a database system performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 11, the database system of claim 10, Cochrane does not explicitly disclose but in the same field of endeavor (database analytical processing systems) Bellamkonda

Bellamkonda discloses wherein the query specifies a group- by operation on grouping sets S, and the view contains result sets for grouping sets C, the controller to determine whether S is a subset of C to determine whether the answer for the query can be satisfied from the view (column 17, lines 58-61; a determination to decide if more operations need to be done to generate result sets satisfy a query is the same as determining whether a group by query specifies grouping sets that are a subset of generated result sets. The result set necessarily created by the disclosed query is the equivalent of S, and records retrieved from the previous rollup operator are the equivalent of C). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of determining whether an answer for the query specifying the group- by operation can be satisfied from the view with Cochrane's teachings of a database system with multiple views with dependencies to obtain a database system performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 12, the database system of claim 11, wherein the controller is adapted to modify a WHERE clause of the query in response to determining that S is a subset of C (element 118, paragraph 0024; the database manager supports the functions of the SQL language including WHERE, and upon determining that S is a subset of C, it was well known at the time the invention was made to modify a query based on predetermined information to obtain results with better performance (paragraph 0007)).

Regarding claim 17, Cochrane does not explicitly disclose wherein updating the first result set and second result set are performed in parallel by the plural access modules. In the same field of endeavor (database analytical processing systems) Bellamkonda discloses wherein updating the first result set and second result set are performed in parallel by the plural access modules (column 16, line 65-column 17, line 1); slaves in Bellamkonda are the equivalent of access modules. The locations where rows are stored that each slave works on are the equivalent of plural storage modules.). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of processing of plural access modules and plural storage modules with Cochrane's teachings of a database system with multiple views with dependencies to obtain a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, lines a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 18, Cochrane does not explicitly disclose further comprising distributing rows of the first and second result sets across the plural access modules.

In the same field of endeavor (database analytical processing systems) Bellamkonda discloses wherein the controller is adapted to distribute rows in the first result set across the access modules of the second grouping set (column 16, line 65-column 17, line 1). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of processing of plural access modules and plural storage modules with Cochrane's teachings of a database system with multiple views with dependencies to obtain a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 19, Cochrane does not explicitly disclose but in the same field of endeavor Bellamkonda discloses wherein the first result set corresponds to a group-by on a first grouping set having N columns, and the second result set corresponds to a group-by on a second grouping set having N-1 columns, and wherein distributing the first result set to compute the second result set comprises distributing the first result set to compute the second result set comprises distributing the first result set based on a hash of the N columns (column 17, lines 25-26), with the column in the first grouping set not present in the second grouping set assigned a predefined value (column 12, line 16-column 12, line 28; Bellamkonda discloses a rollup operation and further discloses a single field (the equivalent of a column) that is aggregated in the grouping set of the calculations of the next stage (the equivalent of the second grouping set). The result table at each stage is the equivalent of a result set. In rollup operations each stage has one less column than the previous stage, with the single column at each stage being set to a predefined value to indicate aggregation. This meets the limitation of the first result set having N columns and the second result set having N-1 columns). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of group-bys with n and n-1 columns, processing of hashing on columns, including an extra column not present in a second grouping set with Cochrane's teachings of a database system with multiple views with dependencies on other views or tables to obtain a system for updating views that processes an extra column in the hashing of a first grouping set than in a second grouping set for the purpose of a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 20, the method of claim 19, further comprising: Cochrane discloses updating a third result set by computing a change to the third result set based on the change to the second result set, wherein the third result set corresponds to a group-by on a third grouping set (paragraph 0035; in this paragraph Cochrane discloses updates to a plurality of views, some of which are computed based on changes to results sets they depend on. Since grouping sets can have any number of grouping attributes specified, the views disclosed in paragraph 0035 that can be derived from grouping sets meet the limitations of updating a third result set by computing a change to the third result set, wherein the third result set corresponds to a group-by on a third grouping set.)

Cochrane does not explicitly disclose but in the same field of endeavor Bellamkonda discloses wherein the third grouping set has N-2 columns and wherein distributing the second result set across the access modules to compute the third result set is based on a hash of the N columns (column 17, lines 25-26), with the columns in the first grouping set not appearing in the third grouping set each assigned to the predefined value (column 12, line 16-column 12, line 28; Bellamkonda discloses a rollup operation and further discloses a single field (the equivalent of a column) that is aggregated in the grouping set of the calculations of the next stage (the equivalent of the second grouping set). The result table at each stage is the equivalent of a result set. In rollup operations each stage has one less column than the previous stage, with the single column at each stage being set to a predefined value to indicate aggregation. Therefore, the third stage, the equivalent of the third result set would have the equivalent of N-2 columns).

Regarding claim 21, the method of claim 20, wherein storing the view comprises storing a view for a cube operation based on a cube function of the N columns (paragraph 0038).

Regarding claim 25, the database system of claim 1, Cochrane does not explicitly disclose wherein processing is done via plural access modules and plural storage modules, the access modules to enable parallel access of data in the plural storage modules.

In the same field of endeavor (database analytical processing systems) Bellamkonda discloses plural access modules and plural storage modules, the access modules to enable parallel access of data in the plural storage modules (column 16, line 65-column 17, line 1); slaves in Bellamkonda are the equivalent of access modules. The locations where rows are stored that each slave works on are the equivalent of plural storage modules.).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of processing of plural access modules and plural storage modules with Cochrane's teachings of a database system with multiple views with dependencies to obtain a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 26, Cochrane does not explicitly disclose further comprising distributing rows of the first and second result sets across the plural access modules.

In the same field of endeavor (database analytical processing systems) Bellamkonda discloses wherein the controller is adapted to distribute rows in the first result set across the access modules of the second grouping set (column 16, line 65-column 17, line 1). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of processing of plural access modules and plural storage modules with Cochrane's teachings of a database system with multiple views with dependencies to obtain a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 27, Cochrane does not explicitly disclose but in the same field of endeavor Bellamkonda discloses wherein the first result set corresponds to a group-by on a first grouping set having N columns, and the second result set corresponds to a group-by on a second grouping set having N-1 columns, and wherein distributing the first result set to compute the second result set comprises distributing the first result set to compute the second result set comprises distributing the first result set based on a hash of the N columns (column 17, lines 25-26), with the column in the first grouping set not present in the second grouping set assigned a predefined value (column 12, line 16-column 12, line 28; Bellamkonda discloses a rollup operation and further discloses a single field (the equivalent of a column) that is aggregated in the grouping set of the calculations of the next stage (the equivalent of the second grouping set). The result table at each stage is the equivalent of a result set. In rollup operations each stage has one less column than the previous stage, with the single column at each stage being set to a predefined value to indicate aggregation. This meets the limitation of the first result set having N columns and the second result set having N-1 columns). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of group-bys with n and n-1 columns, processing of hashing on columns, including an extra column not present in a second grouping set with Cochrane's teachings of a database system with multiple views with dependencies on other views or tables to obtain a system for updating views that processes an extra column in the hashing of a first grouping set than in a second grouping set for the purpose of a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes.

Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 28, the article of claim 27, wherein storing the view comprises storing a view for a cube operation based on a cube function of the N columns (paragraph 0038).

Response to Arguments

Applicant's arguments filed 9/19/2006 have been fully considered but they are not persuasive.

The Applicants argue on page 9 of the remarks that the controller of the Cochrane reference differs from the controller of the instant application in claim 1. Furthermore on page 10, the Applicants argue this difference is because Cochrane, in response to the at least one base table, updates the first result set by computing a change to the first result set based on a change in the at least one base table. On the contrary, the instant application updates a second result set by computing a change to the second result set based on the change to the first result set. The Examiner respectfully disagrees as the invention of Cochrane still anticipates at least these limitations of the Applicants'.

The Examiner agrees with the Applicants that Cochrane's paragraph [0035] states that a self-maintainable materialized view can be maintained incrementally by processing only the set of update rows without reference to the underlying tables of the materialized view query. However, the Examiner submits that this is only a certain class of materialized views. Cochrane states in the next sentence that other materialized views can be maintained incrementally by processing the updated rows and visiting the underlying tables to recompute some portions of the materialized view query. Therefore, the underlying tables have an impact (with respect to updates) to materialized views.

Furthermore, to specify the impact of the underlying tables, Cochrane draws figure 6 to illustrate the incremental maintenance of their invention (description thereof found on page 6). Therein, in step I, a delta is computed as the result of an operation on a base table (see also paragraph [0093]). In paragraph [0104], Cochrane then teaches in the second step (Step II) that the delta from Step I is used to appropriately reflect the changes, which have to be made to the summary table:

Therefore, in summary, Cochrane teaches that underlying tables can be visited for recomputation which is the same as computing a change to a second result set based on the change to the first result set. The computed deltas of steps I and II of paragraphs [0093] and [0104] respectively teach how the changes are reflected.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert M. Timblin whose telephone number is 571-272-5627. The examiner can normally be reached on M-F 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John R. Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Robert M. Timblin
Patent Examiner AU 2167
12/12/2006

ALFORD KINDRED
PRIMARY EXAMINER